TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

KHOA ĐIỆN TỬ

BỘ MÔN: TIN HỌC CÔNG NGHIỆP

BÁO CÁO BÀI TẬP LỚN

MÔN HỌC

Môn	**: Lập trình Python**
Giảng viên hướng dẫn	**: Đỗ Duy Cốp**
Họ và tên sinh viên	**: Vũ Ngọc Bình**
Ngành học	**: Kỹ thuật máy tính**
Lớp	**: K56KMT.01**

THÁI NGUYÊN – 2024

TRƯỜNG ĐHKTCN **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**

KHOA ĐIỆN TỬ ***Độc lập - Tự do - Hạnh phúc***

NHIỆM VU BÀI TẬP LỚN LẬP TRÌNH PYTHON

BỘ MÔN: TIN HỌC CÔNG NGHIỆP

Sinh viên: ***Vũ Ngọc Bình*** *MSSV:****K205480106040***

. Lớp: ***K56KMT***

MSSV: ***K205480106040***

Giáo viên hướng dẫn: Đỗ Duy Cốp

1.Tên đề tài : Gợi ý Anime theo sở thích người xem.....

.....

.....

.....

2. Yêu cầu:

- ✓ Crawl dữ liệu thực trên web dữ liệu anime.
- ✓ Đọc và xử lý được dữ liệu tải về.
- ✓ Nhập vào tên và thể loại, chương trình sẽ gợi ý các phim có nội dung tương tự
- ✓ Yêu cầu có giao diện người dùng (web, app,…)

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

Thái Nguyên, ngày.….tháng…..năm 2023

GIÁO VIÊN HƯỚNG DẪN

(Ký ghi rõ họ tên)

NHẬN XÉT CỦA GIÁO VIÊN CHẤM

.....

.....

.....

.....

.....

Thái Nguyên, ngày.….tháng…..năm 2023

GIÁO VIÊN CHẤM

(Ký ghi rõ họ tên)

CÁC HÌNH ẢNH SỬ DỤNG TRONG ĐỒ ÁN

MỤC LỤC

LỜI MỞ ĐẦU

Với sự phát triển của khoa học công nghệ, phim ảnh ngày nay là một hình thức giải trí phổ biến với mọi người. Phim có đa dạng mọi thể loại tùy vào sở thích người xem, đáp ứng được mọi nhu cầu xem dù là khó khăn nhất. Có người thích xem những phim hành động máu lửa, kích thích mạnh các giác quan người xem, lại có người thích những phim nhẹ nhàng hơn về tình cảm học đường, về tuổi mới lớn… Không chỉ là một hình thức giải trí, phim ảnh còn là cách chúng ta nhìn cuộc sống với một góc nhìn mới, rất thú vị và đầy tính nghệ thuật.

Bên cạnh những bộ phim được vào vai bởi người thật, người ta còn yêu thích những bộ phim hoạt hình được vẽ tỉ mỉ đến từng chi tiết. Phim hoạt hình hiện nay đã không còn chỉ dành cho trẻ em, mà thậm chí được người lớn vô cùng yêu thích. Được đầu tư vào cả hình ảnh và nội dung, đưa đến cho người xem trải nghiệm nghe nhìn vô cùng sống động mà khó phim người thật đóng nào có thể sánh bằng. Ngành công nghiệp phim hoạt hình là ngành công nghiệp trẻ, nhưng có tiềm năng rất lớn sánh ngang với các ngành công nghiệp tỷ đô khác. Nhật Bản đang là quốc gia phát triển rất mạnh về ngành này. Những bộ phim hoạt hình đến từ đất nước mặt trời mọc được rất nhiều người yêu thích vì phong cách nghệ thuật độc đá cùng với sự sáng tạo của các nhà làm phim đã tạo ra một tên gọi riêng cho những bộ phim này – Anime.

Mỗi người đều có một gu thưởng thức phim riêng. Vì thế nên luôn có mọi thể loại phim cho người xem tha hồ lựa chọn. Tuy có đầy đủ tất cả thể loại, nhưng đối với những người mới xem hoặc những người muốn thưởng thức phim theo đúng chủ đề họ muốn, đây vừa là ưu điểm cũng như nhược điểm. Có quá nhiều phim cũng như thể loại khiến cho người mới xem bị ngộp, không thể chọn ra bộ phim mình muốn xem, hoặc những người cần tìm bộ phim có thể loại hoặc chủ đề mong muốn cũng khó có thể tìm ra được "cây kim đáy bể" này. Vì vậy, một hệ thống giúp gợi ý phim theo sở thích xem là rất cần thiết.

Trong phạm vi đồ án này em xin được trình bày quá trình thực hiện bài báo cáo **"Gợi ý Anime theo sở thích người xem"** của Đồ án thực tập chuyên ngành. Cuối cùng, mặc dù đã cố gắng rất nhiều nhưng do thời gian có hạn, khả năng dịch và hiểu tài liệu chưa tốt nên nội dung đồ án này không thể tránh khỏi những thiếu sót, rất mong được sự chỉ bảo, góp ý của các thầy cô và các bạn.

CHƯƠNG I. GIỚI THIỆU CHUNG

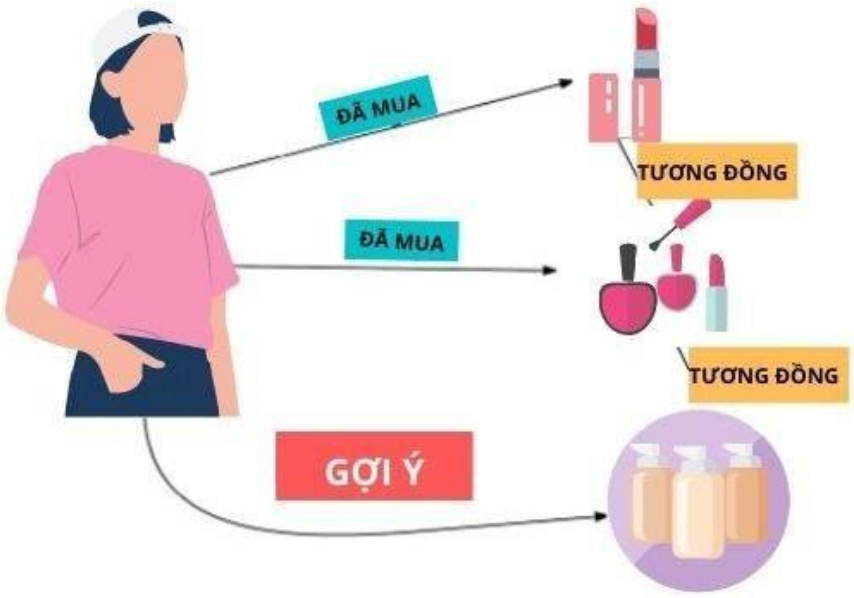
1.1. Hệ thống gợi ý phim

1.1.1. Hệ thống gợi ý là gì ?

Hệ thống gợi ý (Recommender systems hoặc Recommendation systems) là một công nghệ hoặc phần mềm được sử dụng để đề xuất các mục, nội dung hoặc hành động có thể quan tâm hoặc hữu ích cho người dùng. Nó được áp dụng rộng rãi trong nhiều lĩnh vực và ứng dụng khác nhau, từ các nền tảng mua sắm trực tuyến, dịch vụ streaming, tìm kiếm web, mạng xã hội cho đến hệ thống email và ngân hàng trực tuyến.

Hệ thống gợi ý thường dựa trên phân tích dữ liệu và sử dụng các thuật toán máy học hoặc trí tuệ nhân tạo để tìm hiểu về sở thích và hành vi của người dùng. Các thuật toán này phân tích dữ liệu lịch sử, đánh giá, lựa chọn và các yếu tố khác để đưa ra các đề xuất cá nhân hoặc tổng hợp thông tin liên quan.

Qua việc cung cấp các gợi ý cá nhân, hệ thống gợi ý có thể giúp người dùng khám phá thêm thông tin mới, sản phẩm hoặc nội dung mà họ có thể không biết đến. Nó cũng cải thiện trải nghiệm người dùng, giúp tiết kiệm thời gian và tạo ra sự tương tác tốt hơn giữa người dùng và hệ thống.



Hình 1. Ví dụ về hệ thống gợi ý

1.1.2. Hệ thống gợi ý phim là gì ?

Hệ thống gợi ý phim là một dạng hệ thống gợi ý nhằm đề xuất các bộ phim có thể phù hợp với sở thích và sự quan tâm của người dùng. Hệ thống này dựa trên phân tích dữ liệu và sử dụng các thuật toán máy học hoặc trí tuệ nhân tạo để hiểu về lựa chọn, đánh giá và hành vi xem phim của người dùng. Trong dịch vụ xem phim trực tuyến, hệ thống gợi ý có thể đề xuất các bộ phim hoặc chương trình dựa trên sở thích hoặc các nội dung đã được người dùng truy cập.

Hình 2. Netflix – Dịch vụ xem phim trực tuyến trả phí

Hệ thống gợi ý phim có thể sử dụng nhiều phương pháp khác nhau để đưa ra các đề xuất phim cá nhân. Dưới đây là một số phương pháp phổ biến được sử dụng:

- **Dựa trên sở thích cá nhân:** Hệ thống theo dõi lịch sử xem phim của người dùng, đánh giá và đánh dấu các bộ phim mà họ đã xem hoặc yêu thích. Dựa trên thông tin này, hệ thống gợi ý các bộ phim có thể phù hợp với sở thích cá nhân.
- **Dựa trên nội dung:** Hệ thống phân tích nội dung của các bộ phim, bao gồm các yếu tố như thể loại, diễn viên, đạo diễn, câu chuyện, và hợp nhất thông tin này với sở thích cá nhân của người dùng. Dựa trên phân tích nội dung, hệ thống gợi ý các bộ phim có nội dung tương tự hoặc liên quan.

- **Dựa trên thông tin xã hội:** Hệ thống sử dụng thông tin xã hội như đánh giá, bình luận và chia sẻ từ người dùng khác để đưa ra các đề xuất phim. Nếu người dùng có sự tương tác hoặc chia sẻ liên quan đến một bộ phim cụ thể, hệ thống có thể gợi ý các bộ phim tương tự hoặc phổ biến trong cộng đồng.
- **Kết hợp các phương pháp:** Hệ thống gợi ý phim thường kết hợp nhiều phương pháp để cung cấp các đề xuất phim đa dạng và phù hợp với người dùng. Bằng cách kết hợp thông tin về sở thích cá nhân, nội dung phim và thông tin xã hội, hệ thống gợi ý có thể đưa ra các đề xuất tốt hơn.

1.1.3. Ý nghĩa của hệ thống gợi ý phim

Hệ thống gợi ý phim có ý nghĩa quan trọng và nhiều lợi ích đối với người dùng, như sau:

- ✓ **Khám phá phim mới:** Một trong những lợi ích chính của hệ thống gợi ý phim là giúp người dùng khám phá và tiếp cận với các bộ phim mới mà họ có thể chưa biết đến trước đó. Thay vì phải tìm kiếm một cách thủ công, hệ thống gợi ý cung cấp danh sách các bộ phim có thể phù hợp với sở thích và lựa chọn của người dùng, giúp họ mở rộng phạm vi xem phim và khám phá những nội dung mới thú vị.
- ✓ **Tiết kiệm thời gian:** Thay vì phải tự mò mẫm và tìm kiếm một bộ phim để xem, hệ thống gợi ý phim giúp tiết kiệm thời gian cho người dùng. Các đề xuất phim đã được lọc và tùy chỉnh dựa trên sở thích cá nhân, giúp người dùng nhanh chóng tìm thấy các bộ phim có thể phù hợp với họ.
- ✓ **Trải nghiệm cá nhân hóa:** Hệ thống gợi ý phim có khả năng cá nhân hóa đề xuất dựa trên lịch sử xem phim, đánh giá và sở thích của người dùng. Điều này giúp cung cấp trải nghiệm xem phim cá nhân hóa và đáp ứng được mong đợi và yêu cầu riêng của từng người dùng.
- ✓ **Đa dạng hóa nội dung:** Hệ thống gợi ý phim đảm bảo rằng người dùng được tiếp cận với đa dạng nội dung phim. Bằng cách đề xuất các bộ phim từ các thể loại, đạo diễn, diễn viên và quốc gia khác nhau, hệ thống giúp mở rộng sự đa dạng và cung cấp nhiều lựa chọn cho người dùng.
- ✓ **Tăng cường trải nghiệm người dùng:** Hệ thống gợi ý phim cung cấp một trải nghiệm người dùng tốt hơn và nâng cao sự tương tác giữa người dùng và nền tảng xem phim. Bằng cách đưa ra các đề xuất phim chính xác và phù hợp, hệ thống giúp người dùng tìm kiếm, chọn lựa và xem phim một cách dễ dàng và thú vị hơn.

1.1.4. Thuật toán được sử dụng trong đồ án

Thuật toán được sử dụng trong bài là **"Tính toán sự tương đồng giữa 2 vector dựa trên góc cosine"**. Khoảng cách cosine được tính bằng cách đo lường cosine của góc giữa hai vector trong không gian nhiều chiều.

Ta sẽ biến đổi và kết hợp các trường được sử dụng trong dữ liệu như tên phim, thể loại của phim, số tập phim,… thành một vector đặc trưng cho bộ phim đó. Khi ta nhập thông tin vào hệ thống, thông tin đó sẽ được biến đổi thành một vector, sau đó sẽ mang đi so sánh với các vector đã được biến đổi trong hệ thống để tính toán sự tương đồng. Cuối cùng hệ thống sẽ đề xuất ra những phim có sự tương đồng nhất với thông tin đã nhập.

1.2. Anime

1.2.1. Anime là gì ?

Anime là thuật ngữ dùng để chỉ phim hoạt hình có nguồn gốc từ Nhật Bản. Thuật ngữ "Anime" là viết tắt của từ "Animation" (hoạt hình) trong tiếng Anh và được sử dụng rộng rãi trên toàn thế giới để đề cập đến các loại phim hoạt hình Nhật Bản.

Anime không chỉ dành riêng cho trẻ em, mà cũng có các thể loại và nội dung đa dạng, phù hợp với mọi lứa tuổi và sở thích. Anime có thể bao gồm các thể loại như hành động, phiêu lưu, hài hước, tình cảm, kỳ ảo, khoa học viễn tưởng, viễn tưởng, đời thường, lịch sử, kinh dị và nhiều thể loại khác.

Hình 3. Anime – Những bộ phim hoạt hình sản xuất tại Nhật Bản

Phim hoạt hình Nhật Bản (anime) nổi tiếng trên toàn thế giới và có một cộng đồng người hâm mộ đông đảo. Nhiều anime đã trở thành hiện tượng văn hóa và có sức ảnh hưởng lớn, không chỉ trong lĩnh vực giải trí mà còn trong âm nhạc, thời trang, game và các lĩnh vực khác.

Anime được tạo ra bằng cách sử dụng các kỹ thuật hoạt hình truyền thống hoặc kỹ thuật hoạt hình số, và thường có đặc điểm riêng về nét vẽ, phong cách và biểu đạt. Hiện nay, hãng sản xuất anime nổi tiếng nhất tại Nhật là Ghibli với nhiều bộ phim khá nổi tiếng như: "Spirited Away", "Grave of the Fireflies", "My Neighbor Totoro",… được cả thế giới công nhận.

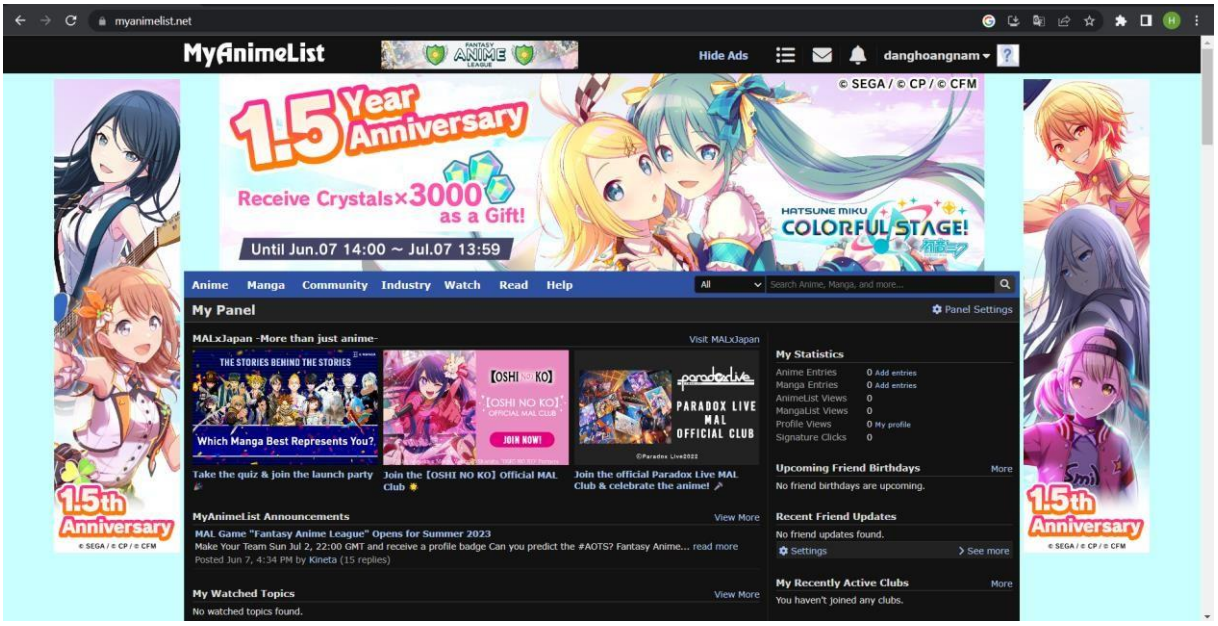
1.2.2. Một số trang web xem Anime trực tuyến

Đây là một số trang web phổ biến để xem anime trực tuyến. Những trang web này được người dùng tin dùng vì độ uy tín và những bộ phim chất lượng cao:

- ✓ **Crunchyroll:** Crunchyroll là một trong những trang web xem anime hàng đầu, cung cấp nhiều bộ anime phổ biến và mới nhất. Trang web này cung cấp cả phiên bản miễn phí và phiên bản trả phí với nội dung chất lượng cao và các tính năng bổ sung.
- ✓ **Funimation:** Funimation là một trang web chuyên về anime, cung cấp nhiều bộ anime đa dạng và phụ đề trong nhiều ngôn ngữ khác nhau. Trang web này cũng có phiên bản miễn phí và phiên bản trả phí với nội dung chất lượng cao.
- ✓ **Netflix:** Netflix cũng là một nền tảng phổ biến để xem anime. Họ cung cấp một loạt các bộ anime phổ biến và độc quyền, cũng như các bộ phim và chương trình truyền hình khác. Netflix yêu cầu một khoản phí hàng tháng để truy cập vào nội dung của họ.
- ✓ **AnimeLab:** AnimeLab là một trang web xem anime phổ biến ở Úc và New Zealand. Họ cung cấp nhiều bộ anime từ các nhà sản xuất hàng đầu và có cả phiên bản miễn phí và phiên bản trả phí.

1.2.3. Trang web MyAnimeList.net

MyAnimeList, thường được viết tắt là **MAL**, là một trang web mạng xã hội và ứng dụng danh mục xã hội về anime và manga. Trang web cung cấp cho người dùng một hệ thống giống như danh sách để sắp xếp và chấm điểm anime và manga. Nó tạo điều kiện cho việc tìm kiếm người dùng chia sẻ thị hiếu tương tự và cung cấp một lượng cơ sở dữ liệu lớn về anime và manga. Trang web tuyên bố có 4,4 triệu mục anime và 775.000 mục manga. Trong năm 2015, trang web đã nhận được 120 triệu khách truy cập mỗi tháng.



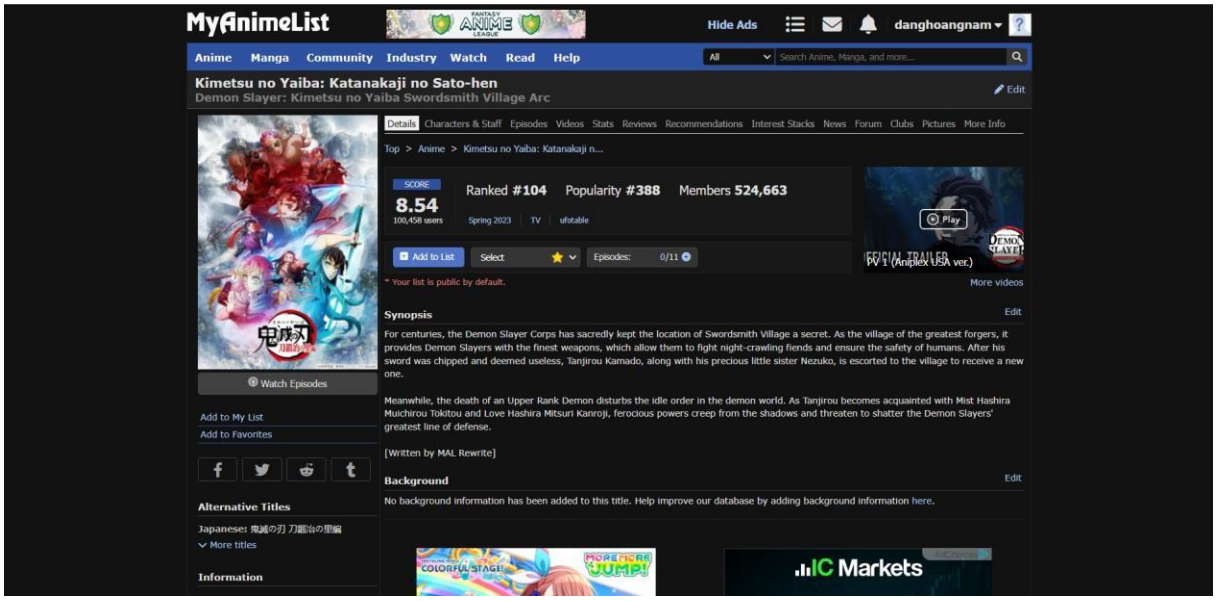
Hình 4. MyAnimeList.net

MyAnimeList cho phép người dùng chấm điểm anime và manga theo thang điểm từ 1 đến 10. Những điểm số này sau đó được tổng hợp để cung cấp cho mỗi chương trình trong cơ sở dữ liệu một thứ hạng từ tốt nhất đến tệ nhất. Thứ hạng của chương trình được tính hai lần một ngày bằng công thức sau:

$$R = \frac{vS + mC}{v + m}$$

Trong đó V là tổng số bình chọn của người dùng, S cho điểm người dùng trung bình, m cho số bình chọn tối thiểu cần có để có được số điểm tính toán (hiện tại là 50), và C cho điểm trung bình trên toàn bộ cơ sở dữ liệu anime/manga.

MyAnimeList liệt kê phim hoạt hình Nhật Bản, phim hoạt hình Hàn Quốc và phim hoạt hình Trung Quốc. Tương tự, MyAnimeList có thông tin về manga, manwha (truyện tranh Hàn Quốc), manhua (truyện tranh Trung Quốc), cũng như dōjinshi (truyện tranh của người hâm mộ) và light novel. Người dùng tạo danh sách mà họ cố gắng hoàn thành. Người dùng có thể gửi đánh giá, viết đề xuất, blog, đăng bài trên diễn đàn của trang web, tạo các câu lạc bộ để hợp nhất với những người có cùng sở thích và đóng góp nguồn cấp tin RSS liên quan đến tin tức về anime và manga.



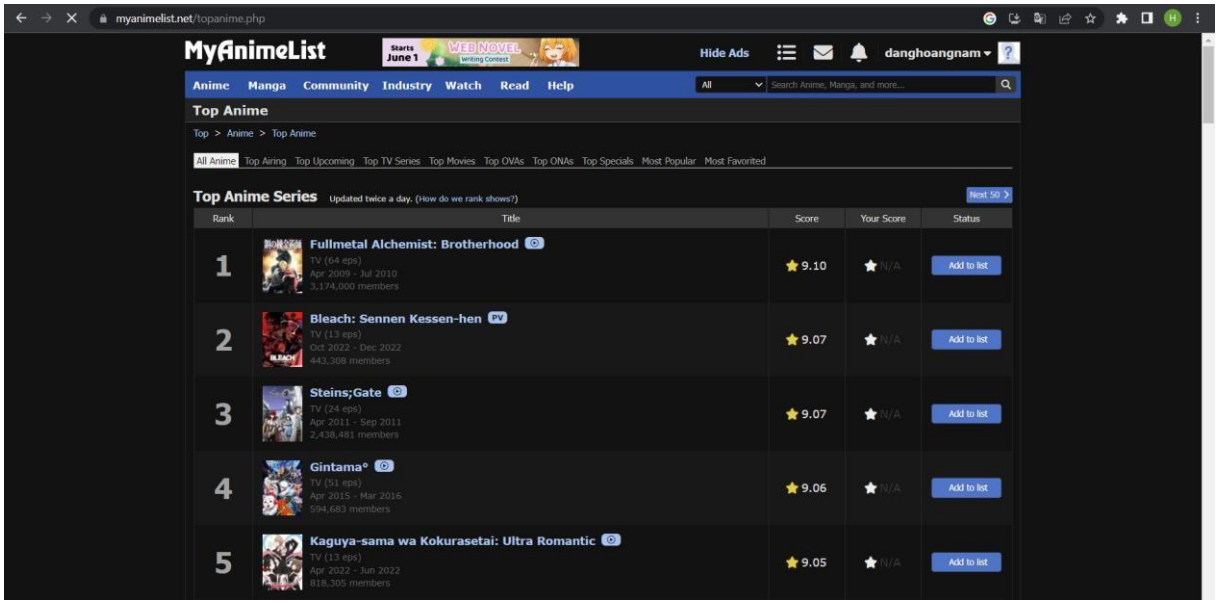
Hình 5. Giao diện chấm điểm, thông tin phim và đánh giá của người xem

CHƯƠNG II. PHÂN TÍCH DỮ LIỆU

2.1. Bộ dữ liệu sử dụng trong đồ án

2.1.1. Giới thiệu về bộ dữ liệu

Bộ dữ liệu sử dụng trong bài tập là bộ dữ liệu được Crawl từ trang web myanimelist.net. Dữ liệu sử dụng thông tin của 2000 bộ anime có số điểm đánh giá cao nhất trên trang web.



Hình 6. Top Anime có điểm đánh giá cao nhất trên MyAnimeList.net

2.1.2. Các trường trong bộ dữ liệu

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Unnamed:	Name	Type	Score	Score Rank	Popularity	Air Date	Studio	Episodes	Genres	Theme	Demographic	
2	0	Fullmetal A	TV	9.14	1	3	Apr 5, 200	Bones	64	Action, Ad	Military	Shounen	
3	1	Spy x Fami	TV	9.09	2	350	Apr 9, 202	Wit Studio	12	Action, Co	Childcare	Shounen	
4	2	Shingeki no	TV	9.08	3	32	Apr 29, 20	Wit Studio	10	Action, Dra	Gore, Milit	Shounen	
5	3	Steins;Gat	TV	9.08	4	13	Apr 6, 201	White Fox	24	Drama, Sci	Psychologi	None	
6	4	GintamaÂ°	TV	9.08	5	335	Apr 8, 201	Bandai Na	51	Action, Co	Gag Humo	Shounen	
7	5	Gintama'	TV	9.05	6	385	Apr 4, 201	Sunrise	51	Action, Co	Gag Humo	Shounen	
8	6	Gintama: T	Movie	9.05	7	1746	8-Jan-21	Bandai Na	1	Action, Co	Gag Humo	Shounen	
9	7	Hunter x H	TV	9.05	8	10	Oct 2, 201	Madhouse	148	Action, Ad	None	Shounen	
10	8	Fruits Bask	TV	9.04	9	551	Apr 6, 202	TMS Enter	13	Drama, Ro	None	Shoujo	
11	9	Gintama':	TV	9.04	10	695	Oct 4, 201	Sunrise	13	Action, Co	Gag Humo	Shounen	
12	10	Ginga Eiyu	OVA	9.03	11	697	Jan 8, 198	K-Factory,	110	Drama, Sci	Adult Cast	None	
13	11	Gintama.	TV	8.99	12	735	Jan 9, 201	Bandai Na	12	Action, Co	Gag Humo	Shounen	
14	12	Kaguya-sa	TV	8.97	13	433	Apr 9, 202	A-1 Picture	0	Comedy	Psychologi	Seinen	
15	13	3-gatsu no	TV	8.96	14	535	Oct 14, 20	Shaft	22	Drama, Sli	Childcare,	Seinen	
16	14	Koe no Kat	Movie	8.96	15	19	########	Kyoto Anir	1	Drama	Romantic	Shounen	
17	15	Gintama	TV	8.95	16	127	Apr 4, 200	Sunrise	201	Action, Co	Gag Humo	Shounen	

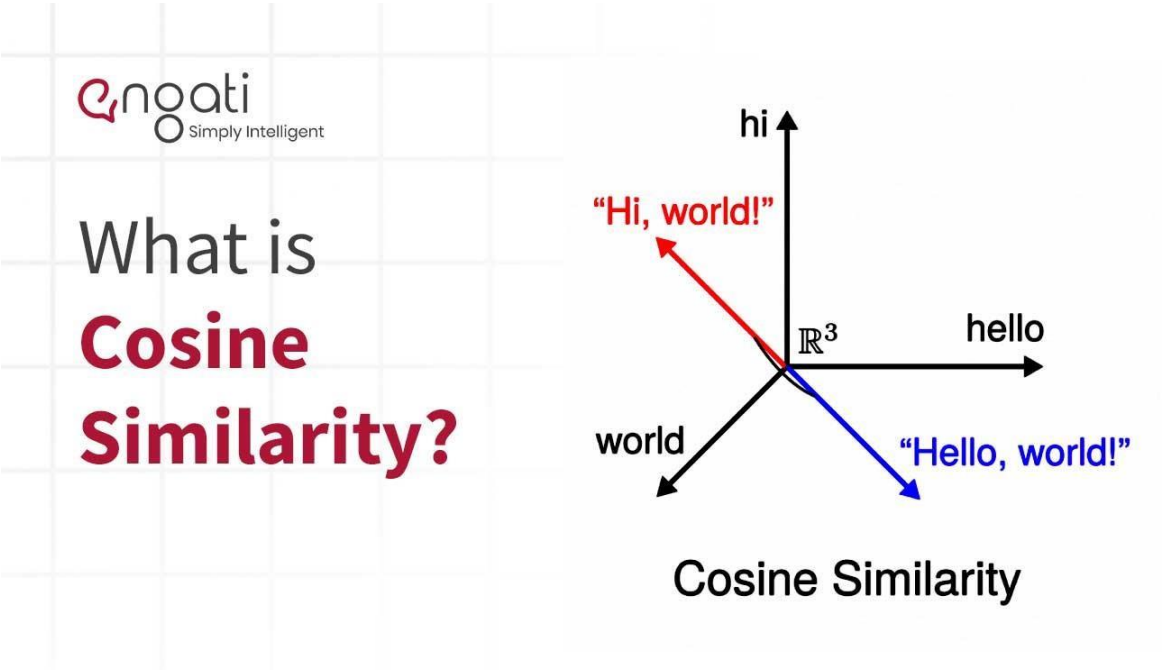
Hình 7. Bộ dữ liệu sau khi Crawl từ trang web

- **Name:** Tên của phim
- **Type:** Hình thức công chiếu của phim (Chiếu TV, chiếu rạp, phần phim mở rộng,…)
- **Score:** Số điểm đánh giá của phim trên thang điểm 10
- **Score Rank:** Xếp hạng số điểm đánh giá của phim
- **Popularity:** Xếp hạng độ phổ biến của phim dựa trên cộng đồng
- **Air Date:** Ngày phát hành và ngày kết thúc phim
- **Studio:** Hãng phim
- **Episodes:** Số tập của phim
- **Genres:** Thể loại của bộ phim (Hài, Hành động, Lãng mạn, Kinh dị, …)
- **Theme:** Chủ đề của phim (Máu me, bạo lực, tâm lý,…)
- **Demographic:** Đối tượng phim hướng đến (Con trai, con gái, …)

2.2. Thuật toán tính toán khoảng cách cosine

2.2.1. Khái niệm

Thuật toán tính toán khoảng cách cosine (Cosine Similarity) là một phương pháp đo đạc mức độ tương đồng giữa hai vectơ trong không gian đa chiều. Nó được sử dụng rộng rãi trong lĩnh vực xử lý ngôn ngữ tự nhiên và khai phá dữ liệu, thường được sử dụng để so sánh sự tương đồng giữa các văn bản, từ ngữ hoặc vectơ biểu diễn của đối tượng trong không gian đa chiều.



Hình 8. Thuật toán Cosine Similarity

Để tính toán khoảng cách cosine giữa hai vectơ, ta thực hiện các bước sau:

1. **Chuẩn hóa các vectơ:** Đầu tiên, chúng ta chuẩn hóa các vectơ đầu vào để chúng có cùng độ dài hoặc độ dài tương đương. Điều này đảm bảo rằng chỉ số cosine similarity sẽ nằm trong khoảng [-1, 1]. Để chuẩn hóa một vectơ, ta chia nó cho độ dài của vectơ đó. Độ dài của một vectơ có thể được tính bằng cách lấy căn bậc hai của tổng bình phương các thành phần của vectơ.

2. **Tính tích vô hướng:** Tiếp theo, ta tính tích vô hướng (dot product) của hai vectơ đã được chuẩn hóa. Tích vô hướng của hai vectơ a và b được tính bằng cách lấy tổng của tích các thành phần tương ứng của hai vectơ đó.

3. **Tính khoảng cách cosine:** Cuối cùng, ta tính khoảng cách cosine bằng cách chia tích vô hướng cho tích của độ dài hai vectơ. Khoảng cách cosine giữa hai vectơ a và b được tính bằng công thức sau:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}},$$

Kết quả là một số trong khoảng [-1, 1], với 1 đại diện cho hai vectơ hoàn toàn tương đồng, 0 đại diện cho hai vectơ không tương đồng và -1 đại diện cho hai vectơ hoàn toàn đối ngược nhau.

2.2.2. Ý nghĩa của thuật toán Cosine Similarity

Thuật toán Cosine Similarity có nhiều ứng dụng trong các lĩnh vực khác nhau. Dưới đây là một số ví dụ phổ biến về ứng dụng của thuật toán cosine similarity:

- **Xử lý ngôn ngữ tự nhiên:** Trong xử lý ngôn ngữ tự nhiên, cosine similarity được sử dụng để đo đạc mức độ tương đồng giữa các văn bản. Nó có thể được áp dụng để tìm kiếm văn bản tương tự, phân loại văn bản, gom cụm văn bản, hoặc xây dựng các hệ thống gợi ý dựa trên nội dung.
- **Hệ thống gợi ý:** Thuật toán cosine similarity được sử dụng trong hệ thống gợi ý để tìm các mục tương tự hoặc người dùng tương đồng. Ví dụ, trên các trang thương mại điện tử, nó có thể được sử dụng để gợi ý sản phẩm tương tự cho người dùng dựa trên lịch sử mua hàng của họ hoặc dựa trên sự tương đồng về mặt nội dung.

- **Phân cụm dữ liệu:** Cosine similarity cũng được sử dụng trong phân cụm dữ liệu để nhóm các điểm dữ liệu có đặc trưng tương tự. Thuật toán có thể giúp xác định sự tương đồng giữa các đối tượng và gom nhóm chúng lại với nhau.
- **Hệ thống lọc cộng tác:** Trong hệ thống lọc cộng tác, cosine similarity có thể được sử dụng để tính toán sự tương đồng giữa các người dùng hoặc các mục tiêu. Kết quả này sau đó có thể được sử dụng để tạo ra các gợi ý cá nhân hóa cho người dùng hoặc tìm ra các người dùng tương tự.
- **Trích xuất thông tin:** Cosine similarity cũng có thể được sử dụng để trích xuất thông tin từ các văn bản. Nó có thể đo đạc mức độ tương tự giữa các văn bản và từ đó xác định được sự liên quan và tính chất của các đối tượng trong văn bản.

2.3. Thuật toán CountVectorizer

2.3.1. Khái niệm

Thuật toán CountVectorizer là một phương pháp trong xử lý ngôn ngữ tự nhiên để biểu diễn văn bản thành vectơ đặc trưng dựa trên tần số xuất hiện của các từ trong văn bản. Nó là một phần của thư viện scikit-learn (sklearn) của Python và được sử dụng rộng rãi trong các tác vụ như phân loại văn bản, gom cụm và trích xuất thông tin.

Cách hoạt động của CountVectorizer:

1. **Chia tách văn bản:** Đầu tiên, CountVectorizer sẽ chia tách các văn bản thành các "token" (có thể là từ, ký tự, hoặc n-gram, phụ thuộc vào cấu hình).
2. **Xây dựng từ điển:** CountVectorizer xây dựng một từ điển của các từ duy nhất trong tập dữ liệu. Mỗi từ sẽ được gán một chỉ mục dcuy nhất.
3. **Tính toán tần số xuất hiện:** CountVectorizer sử dụng từ điển để tính toán tần số xuất hiện (số lần xuất hiện) của mỗi từ trong mỗi văn bản.
4. **Biểu diễn thành vector:** Kết quả là một ma trận mà mỗi hàng đại diện cho một văn bản và mỗi cột đại diện cho tần số xuất hiện của một từ trong văn bản tương ứng.

CountVectorizer là một công cụ mạnh mẽ để biểu diễn văn bản thành dạng số học, tuy nhiên, nó không lưu trữ thông tin về tần số xuất hiện tương đối giữa các văn bản và không xem xét các khía cạnh ngữ nghĩa của từ.

2.3.2. Ví dụ về CountVectorizer

Xem xét một vài văn bản mẫu từ một tài liệu (mỗi văn bản là một phần tử danh sách):

document = [“One Geek helps Two Geeks”, “Two Geeks help Four Geeks”, “Each Geek helps many other Geeks at GeeksforGeeks.”]

CountVectorizer tạo một ma trận (matrix) trong đó mỗi từ duy nhất được biểu thị bằng một cột (column) của ma trận và mỗi mẫu văn bản từ tài liệu là một hàng (row) trong ma trận. Giá trị của mỗi ô (cell) là số lượng từ trong mẫu văn bản cụ thể đó:

	at	each	four	geek	geeks	geeksforgeeks	help	helps	many	one	other	two
document[0]	0	0	0	1	1	0	0	1	0	0	0	1
document[1]	0	0	1	0	2	0	1	0	0	0	0	1
document[2]	1	1	0	1	1	1	0	1	1	0	1	0

- ✓ Có 12 từ duy nhất trong tài liệu, được biểu diễn dưới dạng các cột của bảng.
- ✓ Có 3 mẫu văn bản trong tài liệu, mỗi mẫu được biểu thị dưới dạng các hàng của bảng.
- ✓ Mỗi ô chứa một số, đại diện cho số lượng từ trong văn bản cụ thể đó.
- ✓ Tất cả các từ đã được chuyển đổi thành chữ thường.
- ✓ Các từ trong các cột đã được sắp xếp theo thứ tự bảng chữ cái.

Bên trong CountVectorizer, những từ này không được lưu trữ dưới dạng chuỗi. Thay vào đó, chúng được cung cấp một giá trị chỉ số (index) cụ thể. Trong trường hợp này, "at" sẽ có chỉ số 0, "each" sẽ có chỉ số 1, "four" sẽ có chỉ số 2, v.v. Biểu diễn thực tế đã được hiển thị trong bảng dưới đây.

0	1	2	3	4	5	6	7	8	9	10	11
0	0	0	1	1	0	0	1	0	0	0	1
0	0	1	0	2	0	1	0	0	0	0	1
1	1	0	1	1	1	0	1	1	0	1	0

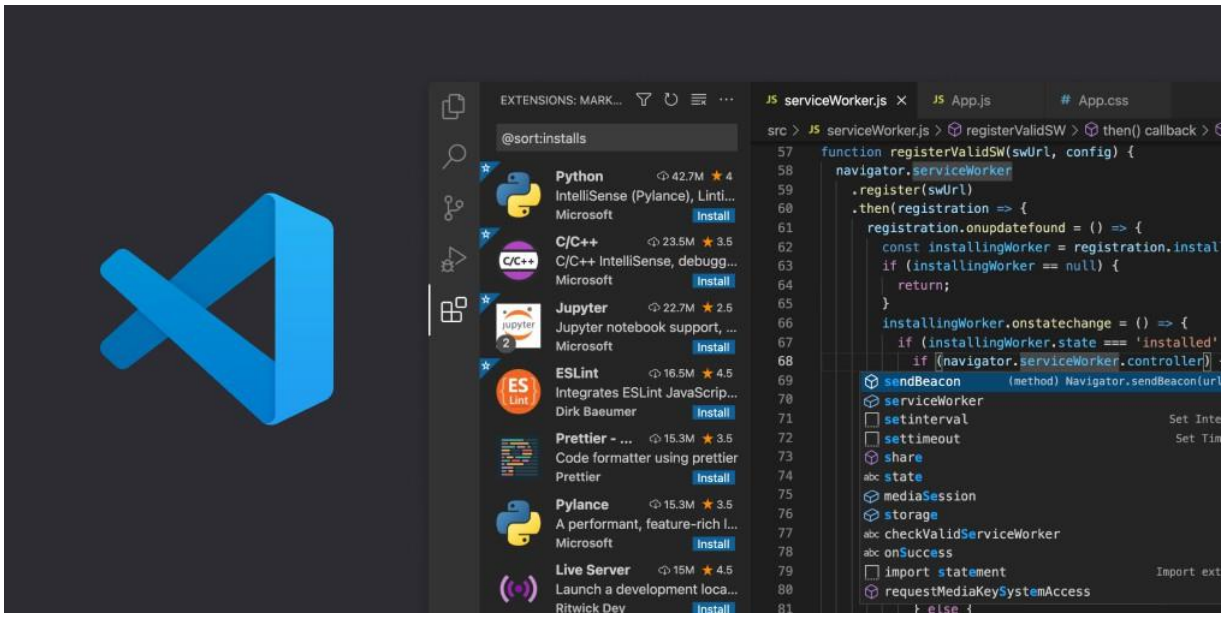
CHƯƠNG III. THIẾT KẾ CHƯƠNG TRÌNH

3.1. Ngôn ngữ lập trình và môi trường làm việc

3.1.1. Visual Studio Code

Visual Studio Code là một trình soạn thảo mã nguồn được phát triển bởi Microsoft dành cho Windows, Linux và macOS. Nó hỗ trợ chức năng debug, đi kèm với Git, có chức năng nổi bật cú pháp (syntax highlighting), tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Nó cũng cho phép tùy chỉnh, do đó, người dùng có thể thay đổi theme, phím tắt, và các tùy chọn khác. Nó miễn phí và là phần mềm mã nguồn mở theo giấy phép MIT, mặc dù bản phát hành của Microsoft là theo giấy phép phần mềm miễn phí.

Visual Studio Code được dựa trên Electron, một nền tảng được sử dụng để triển khai các ứng dụng Node.js máy tính cá nhân chạy trên động cơ bố trí Blink. Mặc dù nó sử dụng nền tảng Electron nhưng phần mềm này không phải là một bản khác của Atom, nó thực ra được dựa trên trình biên tập của Visual Studio Online (tên mã là "Monaco").



Hình 9. Logo Visual Studio Code và giao diện làm việc

Visual Studio Code có thể được mở rộng qua plugin. Điều này giúp bổ sung thêm chức năng cho trình biên tập và hỗ trợ thêm ngôn ngữ. Một tính năng đáng chú ý là khả năng tạo phần mở rộng để phân tích mã, như là các linter và công cụ phân tích, sử dụng Language Server Protocol.

Visual Studio Code là một trình biên tập mã. Nó hỗ trợ nhiều ngôn ngữ và chức năng tùy vào ngôn ngữ sử dụng theo như trong bảng sau. Nhiều chức năng của Visual Studio Code không hiển thị ra trong các menu tùy chọn hay giao diện người dùng. Thay vào đó, chúng được gọi thông qua khung nhập lệnh hoặc qua một tập tin .json (ví dụ như tập tin tùy chỉnh của người dùng). Khung nhập lệnh là một giao diện theo dòng lệnh. Tuy nhiên, nó biến mất khi người dùng nhấp bất cứ nơi nào khác, hoặc nhấn tổ hợp phím để tương tác với một cái gì đó ở bên ngoài đó. Tương tự như vậy với những dòng lệnh tốn nhiều thời gian để xử lý. Khi thực hiện những điều trên thì quá trình xử lý dòng lệnh đó sẽ bị hủy.

3.1.2. Python

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học; được dùng rộng rãi trong phát triển trí tuệ nhân tạo. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.

Python luôn được xếp hạng vào những ngôn ngữ lập trình phổ biến nhất.

Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

Hình 10. Ngôn ngữ lập trình Python

Python là một ngôn ngữ lập trình đa mẫu hình. Lập trình hướng đối tượng và lập trình cấu trúc được hỗ trợ hoàn toàn, và nhiều tính năng của nó cũng hỗ trợ lập trình hàm và lập trình hướng khía cạnh (bao gồm siêu lập trình và siêu đối tượng. Các mẫu hình khác cũng được hỗ trợ thông qua các phần mở rộng, bao gồm thiết kế theo hợp đồng và lập trình logic.

Python sử dụng kiểu động và một dạng kết hợp giữa đếm tham chiếu và bộ dọn rác kiểm tra theo chu kì để quản lí bộ nhớ. Nó cũng có tính năng phân giải tên động (liên kết muộn), cho phép liên kết các tên biến và phương thức trong quá trình thực thi chương trình.

Triết lý căn bản của ngôn ngữ Python được trình bày trong tài liệu *The Zen of Python* (*PEP 20*), có dạng thơ Haiku, tóm gọn như sau:

- Đẹp đẽ tốt hơn xấu xí
- Minh bạch tốt hơn ngầm định
- Đơn giản tốt hơn phức tạp
- Phức tạp tốt hơn rắc rối
- Tính dễ đọc rất quan trọng.

Thay vì tích hợp hết tất cả các tính năng vào phần cốt lõi, Python được thiết kế để dễ dàng mở rộng (bằng các mô đun). Tính mô đun nhỏ gọn này đã làm cho Python trở nên phổ biến như là một cách thêm các giao diện lập trình được vào các ứng dụng hiện có. Tầm nhìn của Van Rossum về một ngôn ngữ có phần lõi nhỏ với một thứ viện chuẩn rộng lớn và một trình thông dịch dễ dàng mở rộng bắt nguồn từ việc ông nản lòng trước ABC, một ngôn ngữ lập trình tán thành hướng tiếp cận ngược lại. Python thường được mô tả là một ngôn ngữ "tặng kèm pin" nhờ vào thư viện chuẩn bao quát của nó.

Python nỗ lực hướng đến một cú pháp đơn giản hơn, gọn gàng hơn trong khi vẫn cho các nhà phát triển lựa chọn phương pháp viết mã của họ. Đối lập với khẩu hiệu "có nhiều hơn một cách để làm việc này", triết lý thiết kế của Python lại nằm trong châm ngôn "chỉ nên có một— và tốt nhất là chỉ một—cách rõ ràng để làm việc này". Alex Martelli, một Viện sĩ (Fellow) tại Tổ chức Phần mềm Python (Python Software Foundation) và là một tác giả viết sách Python, viết rằng "Mô tả một thứ gì đó là "tài tình" *không* được coi là một lời khen ngợi trong văn hoá Python."

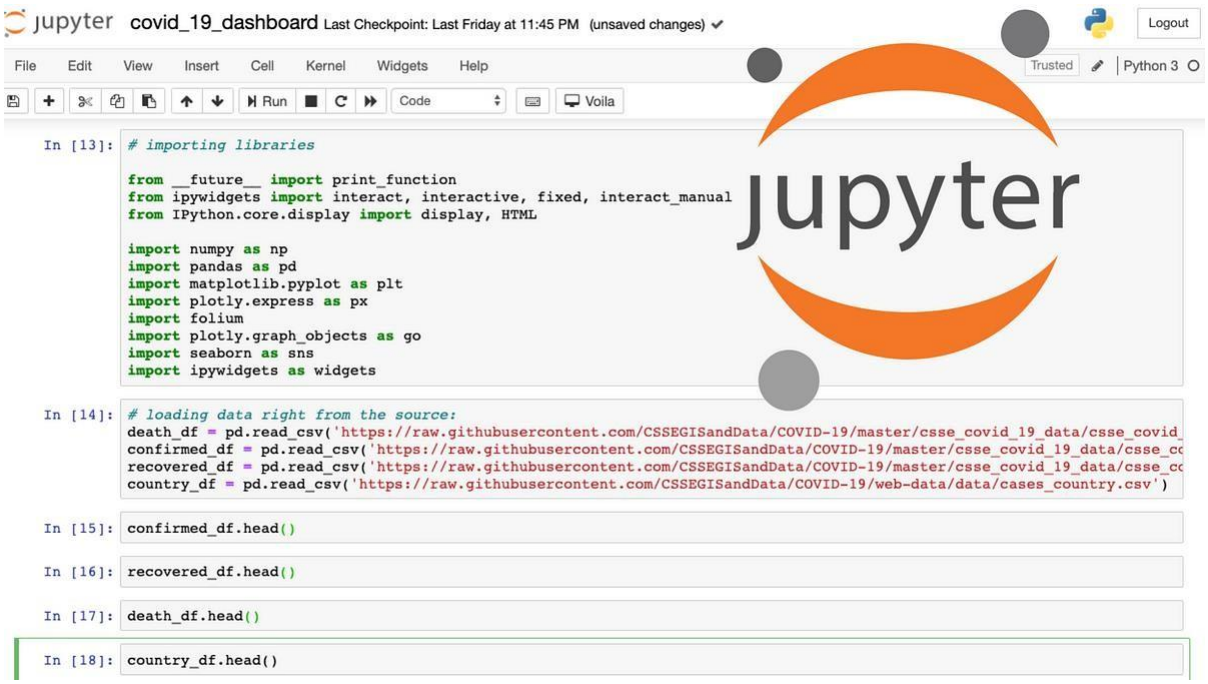
3.1.3. Jupyter Notebook

Jupyter là một thuật ngữ được ghép từ ba ngôn ngữ lập trình Julia, Python và R. Trước đây Jupyter Notebook có tên là IPython Notebook, đến năm 2014 tách ra khỏi IPython và đổi tên thành Jupyter Notebook.

Jupyter Notebook là một nền tảng tính toán khoa học mã nguồn mở, bạn có thể sử dụng để tạo và chia sẻ các tài liệu có chứa code trực tiếp, phương trình, trực quan hóa dữ liệu và văn bản tường thuật.

Jupyter Notebook được coi là môi trường điện toán tương tác đa ngôn ngữ, hỗ trợ hơn 40 ngôn ngữ lập trình cho người dùng.

Jupyter Notebook cho phép bạn viết và thực thi mã trong các ô (cells) riêng lẻ và xem kết quả ngay lập tức. Bạn có thể sử dụng nhiều ngôn ngữ lập trình khác nhau như Python, R, Julia và nhiều ngôn ngữ khác. Điều này rất hữu ích khi bạn muốn thực hiện các tính toán, trực quan hóa dữ liệu, viết báo cáo hoặc chia sẻ ý tưởng của mình.



Hình 11. Cấu trúc chia các ô riêng lẻ của Jupyter Notebook

Jupyter Notebook được viết bằng các ngôn ngữ như Python, R và Julia, nền tảng này hiện đang được sử dụng rộng rãi. Bên cạnh đó, Jupyter còn tạo ra tài liệu, trực quan hóa dữ liệu và lưu trữ chúng một cách dễ dàng hơn rất nhiều. Dưới đây là một số lợi ích mà Jupyter Notebook mang lại:

- **Phân tích khám phá dữ liệu (Exploratory Data Analysis):** Jupyter cho phép người dùng xem kết quả của code in-line (mã inline) mà không cần phụ thuộc vào các phần khác của code. Trong Notebook mọi ô của code có thể được kiểm tra bất cứ lúc nào, điều này đã giúp Jupyter trở nên khác biệt so với các ID như Pycharm, VSCode. Việc Jupyter có thể xuất code in-line đã giúp ích rất nhiều trong quá trình phân tích khám phá dữ liệu (EDA).
- **Bộ đệm dễ dàng trong ô tích hợp:** Từng ô tự duy trì trạng thái hoạt động sẽ hơi khó, nhưng với Jupyter, công việc này sẽ được thực hiện tự động. Vì Jupyter lưu trữ kết quả hoạt động của mọi ô đang chạy, cho dù là code đang đào tạo mô hình machine learning hay code đang tải xuống gigabyte dữ liệu từ một máy chủ từ xa.
- **Độc lập ngôn ngữ:** Jupyter Notebook ở định dạng JSON, vì thế nó được biết đến là một nền tảng độc lập cũng như độc lập về ngôn ngữ.
- **Trực quan hóa dữ liệu (Data Visualisation):** Jupyter Notebook hỗ trợ trực quan hóa dữ liệu và hiển thị thêm một số đồ họa và biểu đồ. Những điều này được tạo ra từ code với sự trợ giúp của các mô -đun như Matplotlib, Plotly hoặc Bokeh. Ngoài ra, Jupyter còn cho phép người dùng cùng chia sẻ code và bộ dữ liệu hoặc thay đổi tương tác với nhau.
- **Tương tác trực tiếp với code:** Jupyter Notebook sử dụng "ipywidgets" packages, cung cấp cho người dùng giao diện chuẩn nhằm khám phá sự tương tác trực tiếp với code và với dữ liệu. Người dùng có thể chỉnh sửa và chạy code, làm cho code của Jupyter non-static. Ngoài ra, nó còn cho phép người dùng kiểm soát nguồn đầu vào của code và phản hồi lại trực tiếp trên trình duyệt.
- **Các mẫu code tài liệu:** Jupyter giúp người dùng dễ dàng giải thích từng dòng code của họ với các phản hồi được đính kèm. Dù trong code đã có đầy đủ các chức năng nhưng người dùng vẫn có thể tăng thêm sự tương tác bằng các lời giải thích.

3.2. Code chương trình

```
import numpy as np
import pandas as pd
from sklearn.feature_extraction import text
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import streamlit as st
import re
import string
```

Đầu tiên ta thêm các thư viện cần thiết cho bài toán như Numpy và Pandas để làm việc với dữ liệu. Thư viện String để xử lý các chuỗi. Trong thư viện Sklearn, ta sử dụng các hàm như text để làm việc với văn bản, hàm CountVectorizer là hàm chuyển đổi văn bản thành vector đặc trưng, hàm cosine_similarity là hàm so sánh khoảng cách cosine. Cuối cùng là thư viện streamlit để xây dựng ứng dụng web app.

```
def clean(text):
    text  =  str(text).lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    return text
```

Bước đầu tiên ta xây dựng một hàm có tên clean(). Hàm này sẽ thực hiện nhiệm vụ lọc các văn bản, loại bỏ các dấu câu, kí tự đặc biệt, biến chữ hoa thành chữ thường.

```
@st.cache(allow_output_mutation=True)
def load_data():
    anime = pd.read_csv("D:\\Do an thuc tap chuyen nganh\\Du lieu
2023\\anime2023.csv")
    anime = anime[['Name', 'Type', 'Score', 'Studio', 'Episodes', 'Genres',
'Theme', 'Demographic']]

    anime["clean_Name"] = anime["Name"].apply(clean)
    anime["clean_Genres"] = anime["Genres"].apply(clean)
    anime["clean_Theme"] = anime["Theme"].apply(clean)
    anime["clean_Studio"] = anime["Studio"].apply(clean)
    anime["clean_Demographic"] = anime["Demographic"].apply(clean)

    anime["clean_Genres_Theme_Demographic"] = anime["clean_Genres"] + " " +
anime["clean_Theme"] + " " + anime["clean_Demographic"]
    anime["clean_Feature"] = anime["clean_Name"] + " " +
anime["clean_Genres_Theme_Demographic"]

    indices =
pd.Series(anime.index,index=anime['clean_Feature']).drop_duplicates()
    return anime, indices
```

Ta sẽ xây dựng hàm load_data(). Hàm này sẽ được sử dụng để tải dữ liệu từ tệp CSV và thực hiện các bước tiền xử lý dữ liệu. Câu lệnh **@st.cache(allow_output_mutation=True)** được sử dụng để đánh dấu hàm load_data() là một hàm được lưu trữ bởi Streamlit. Điều này cho phép Streamlit lưu trữ kết quả của hàm và sử dụng lại chúng giữa các phiên làm việc để tăng tốc độ thực thi và tránh tải lại dữ liệu mỗi lần ứng dụng được khởi động lại. Tham số **allow_output_mutation=True** cho phép các đối tượng có thể thay đổi trong hàm được lưu trữ và tái sử dụng.

Ta đọc dữ liệu từ tệp CSV có đường dẫn trên và lưu nó vào Dataframe có tên "anime". Tiếp theo ta chọn các cột dữ liệu quan trọng để hiển thị ra web app đó là 'Name', 'Type', 'Score', 'Studio', 'Episodes', 'Genres', 'Theme', 'Demographic'. Các cột này sẽ được sử dụng cho việc gợi ý Anime.

Ta áp dụng hàm clean() cho các cột 'Name', 'Studio', 'Genres', 'Theme', 'Demographic' , tạo các cột mới có tên "clean_Name", "clean_Genres", "clean_Studio", "clean_Theme" và "clean_Demographic" chứa các chuỗi đã được lọc. Cuối cùng tạo một cột mới có tên "clean_Feature" là sự kết hợp của 5 chuỗi "clean_Name", "clean_Genres", "clean_Studio", "clean_Theme", "clean_Demographic"". Các chuỗi được cách nhau bởi dấu " ".

Ta tạo một Serries có tên "indices" có công dụng để tìm kiếm nhanh dựa trên cột "clean_Feature" vừa tạo. drop_duplicates() để loại bỏ các thành phần trùng lặp. Có thể hiểu đơn giản là dòng code này sẽ gán một mã số riêng cho mỗi chuỗi của cột "clean_Feature". Mỗi mã số này sẽ là duy nhất, do vậy ta có thể trỏ đến mã số để lấy thông tin của chuỗi. Cuối cùng hàm trả về dataframe "anime" và Serries "indices" để sử dụng trong những đoạn code sau.

```
def get_similarity(title, anime, indices):
    new_title = False
    feature  =  anime["clean_Feature"].tolist()

    if not(title in feature):
        new_title = True
        feature.append(title)

    tfidf  =  text.CountVectorizer()
    tfidf_matrix  =  tfidf.fit_transform(feature)

    similarity  =  cosine_similarity(tfidf_matrix)

    del tfidf
    del tfidf_matrix

    if  new_title:
        del feature
```

```
        return similarity[len(similarity) - 1]
    else:
        del feature
        index = pd.Series(indices[title])
        return similarity[index[0]]
```

Ta xây dựng hàm có tên get_similarity() dùng để tính độ tương đồng giữa anime nhập vào từ bàn phím và danh sách anime có trong dữ liệu. Đầu tiên ta tạo một biến có tên "new_title" có giá trị mặc định là False. Biến này sẽ được dùng để kiểm tra xem anime nhập vào có tồn tại trong dữ liệu hay không.

Tạo một list danh sách có tên "feature" bằng cách lấy dữ liệu của cột "clean_Feature" và chuyển thành danh sách list.

Một hàm if để kiểm tra xem anime nhập vào có tồn tại trong dữ liệu hay không. Nếu không tồn tại, biến "new_title" sẽ chuyển thành giá trị True, sau đó thêm anime vừa nhập vào danh sách "feature" sử dụng "append".

Tạo một đối tượng CountVectorizer có tên "tfidf" để biểu diễn các chuỗi thành ma trận đếm. "tfidf" hay còn gọi là TF-IDF, là một phương pháp phổ biến trong xử lý ngôn ngữ tự nhiên (NLP) được sử dụng để đánh giá độ quan trọng của một từ trong văn bản. Trong phương pháp này, mỗi từ sẽ được đánh giá điểm theo hai yếu tố chính: tần số của từ đó trong văn bản (TF) và tầm quan trong của từ đó trong văn bản (IDF). Tiếp đến ta biến đổi danh sách "feature" thành ma trận đếm sử dụng "fit_transform" của đối tượng CountVectorizer. Đặt tên ma trận vừa tạo là "tfidf_matrix". Cuối cùng tính tính ma trận độ tương đồng Cosine từ ma trận "tfidf_matrix" và đặt tên là "similarity".

Ta sử dụng if-else để lấy độ tương đồng "similarity" vừa tính. Nếu anime vừa nhập không tồn tại trong dữ liệu (new_title = True), anime này sẽ nằm cuối cùng trong ma trận tương đồng "similarity" bằng cách lấy độ dài của "similarity" trừ đi 1.

Nếu anime có tồn tại trong dữ liệu, ta sẽ tìm mã số của anime đó, sử dụng chỉ số "indices". Cuối cùng trả về hàng tương ứng với chỉ số "index[0]" trong "similarity". Đây là chỉ số tương đồng giữa anime vừa nhập và các anime khác trong dữ liệu.

```
def anime_recommendation(name, genres, anime, indices):
    cleaned_name = clean(name)
    cleaned_genres = clean(genres)
    cleaned_nameGenres = cleaned_name + " " + cleaned_genres
```

```
    similarity = get_similarity(cleaned_nameGenres, anime, indices)

    similarity_scores  =  list(enumerate(similarity))

    similarity_scores  =  sorted(similarity_scores, key=lambda x:  x[1],
reverse=True)

    similarity_scores = filter(lambda x: x[1] > 0, similarity_scores)
    similarity_scores  =  list(similarity_scores)

    movieindices = [i[0] for i in similarity_scores if i[0] < len(anime)]

    scores = [i[1] for i in similarity_scores if i[0] < len(anime)]

    result = pd.DataFrame([anime.iloc[i] for i in movieindices])

    result['score'] = scores

    result  =  result['Name'].values.tolist()

    return result[:10]
```

Cuối cùng là xây dựng hàm anime_recommendation() dùng để gợi ý Anime cho người dùng. Hàm này nhận vào tên "name" và thể loại "genres" mà người dùng muốn tìm kiếm. Ta áp dụng hàm clean() cho hai chuỗi vừa nhập để lọc văn bản, sau đó gán hai chuỗi đã sửa vào hai biến "cleaned_name" và "clean_genres". Tạo chuỗi kết hợp giữa hai chuỗi trên, có tên là "cleaned_NameGenres", được cách nhau bởi dấu " ".

Ta gọi hàm "get_similarity" đã tạo bên trên với tên "similarity" để tính độ tương đồng giữa các chuỗi "cleaned_NameGenres" vừa nhập và tất cả các tiêu đề Anime trong dữ liệu. Kết quả được lưu vào biến "similarity" là một mảng 2D chứa các giá trị độ tương đồng.

Tạo một danh sách có tên "similarity_scores" chứa 2 thành phần là chỉ số "indices" và điểm số tương đồng từ "enumerate(similarity)". Ta chọn chỉ số điểm số tương đồng sử dụng hàm "lambda", sử dụng "sorted" để sắp xếp theo giá trị giảm dần. Sử dụng hàm "filter" để chỉ giữ lại các điểm số tương đồng lớn hơn 0

Tạo một biến "movieindices" là một danh sách chứa chỉ số "indices" của anime trong "similarity_score" mà có chỉ số nhỏ hơn độ dài của anime.csv. Tạo biến "scores" là danh sách chưa điểm số tương đồng của anime trong "similarity_score" mà có chỉ số nhỏ hơn độ dài của anime.csv.

Ta tạo một DataFrame mới có tên “result” bằng cách sử dụng vòng lặp for để lấy chỉ số trong “movieindices” và trích xuất hàng tương ứng trong anime.csv. Gán cột “score” cho DataFrame “result” với các giá trị từ danh sách “scores”. Chọn cột “Name” và chuyển đổi thành danh sách list. Cuối cùng hàm trả về 10 anime đầu tiên trong danh sách list “result”.

```
def main():
    st.title("Gợi ý Anime theo sở thích người xem")

    data = load_data()
    anime = data[0]
    indices = data[1]

    name = st.text_input("Nhập tên Anime bạn thích :")
    genres = st.text_input("Nhập thể loại phim bạn thích :")

    if st.button("Gợi ý"):
        with st.spinner(text='Đang xử lý ...'):
            recommendations = anime_recommendation(name, genres, anime,
indices)

            if recommendations:
                # st.success('Hoàn thành')
                st.header("Dưới đây là các bộ Anime gợi ý cho bạn :")

                for i, recommendation in enumerate(recommendations):
                    st.info(f"{i + 1}. {recommendation}")
                    anime_info = anime.loc[anime['Name'] == recommendation]
                    st.write(f"Điểm số đánh giá :
{anime_info['Score'].values[0]}")
                    st.write(f"Thể loại : {anime_info['Genres'].values[0]}")
                    st.write(f"Hình thức : {anime_info['Type'].values[0]}")
                    st.write(f"Số tập phim :
{anime_info['Episodes'].values[0]}")
                    st.write(f"Hãng phim :
{anime_info['Studio'].values[0]}")
                    st.write(f"Chủ đề : {anime_info['Theme'].values[0]}")
                    st.write(f"Đối tượng khán giả :
{anime_info['Demographic'].values[0]}")
                    st.write("-------")

                # for i, recommendation in enumerate(recommendations):
                #     st.write(f"{i + 1}. {recommendation}")
            else:
                st.write("Không có gợi ý nào phù hợp")

if __name__== "__main__":
    main()
```

Ta xây dựng hàm main() để xử lý giao diện người dùng và hiển thị kết quả gợi ý anime dựa trên sở thích người dùng. Ta tạo một tiêu đề cho giao diện Streamlit bằng dòng lệnh st.title("Gợi ý anime theo sở thích người xem"). Ta gọi hàm load_data() bằng lệnh data = load_data() để tải dữ liệu anime.csv và lưu vào biến "data". Biến này chứa 2 phần tử, một là Dataframe "anime" chứa thông tin về Anime, hai là Serries "indices" chứa chỉ số để tra cứu nhanh.

Tiếp theo ta tạo hai ô để nhập văn bản. name = st.text_input("Nhập tên anime:") để tạo một ô nhập tên anime người dùng thích. Tương tự với genres = st.text_input("Nhập thể loại anime:") để tạo một ô nhập vào thể loại phim.

if st.button("Gợi ý") để tạo nút bấm có tên "Gợi ý" và kiểm tra xem người dùng đã bấm nút hay chưa. Nếu đã nhấn, một vòng quay sẽ được hiển thị lên mô tả hệ thống đang xử lý. Ta gọi hàm anime_recommendation() để lấy danh sách anime được gợi ý dựa trên tên và thể loại đã nhập và lưu kết quả vào biến "recommendations".

Tiếp theo chương trình sẽ kiểm tra có anime gợi ý nào hay không. Nếu có, chương trình sẽ hiển thị tiêu đề "Đây là các anime gợi ý cho bạn", sau đó liệt kê các anime gợi ý, kèm theo các thông tin của anime đó như điểm số đánh giá, thể loại, số tập,… Mỗi anime được hiển thị trong một khung thông tin. Nếu không có gợi ý nào phù hợp, chương trình sẽ hiện ra thông báo "Không có gợi ý phù hợp.".

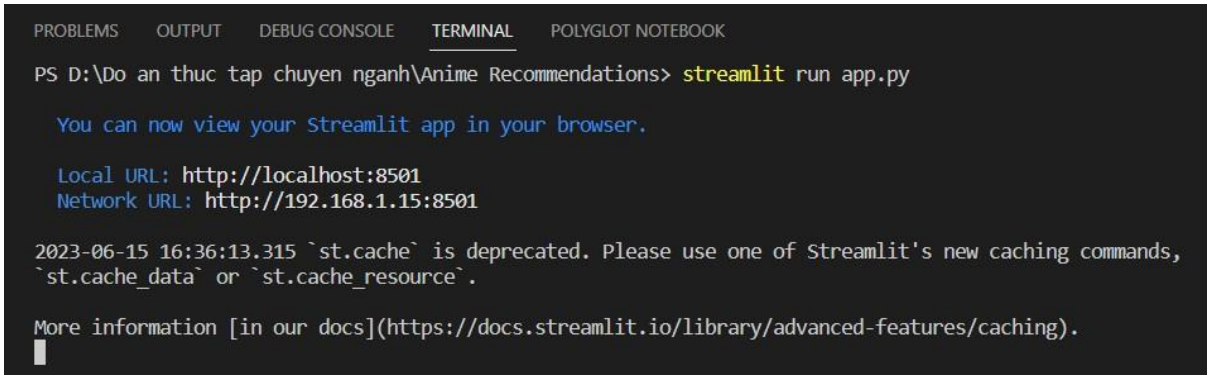
Cuối cùng là dòng if __name__ == "__main__" sẽ đảm bảo chương trình chỉ được chạy khi gọi trực tiếp, không chạy khi import vào module khác.

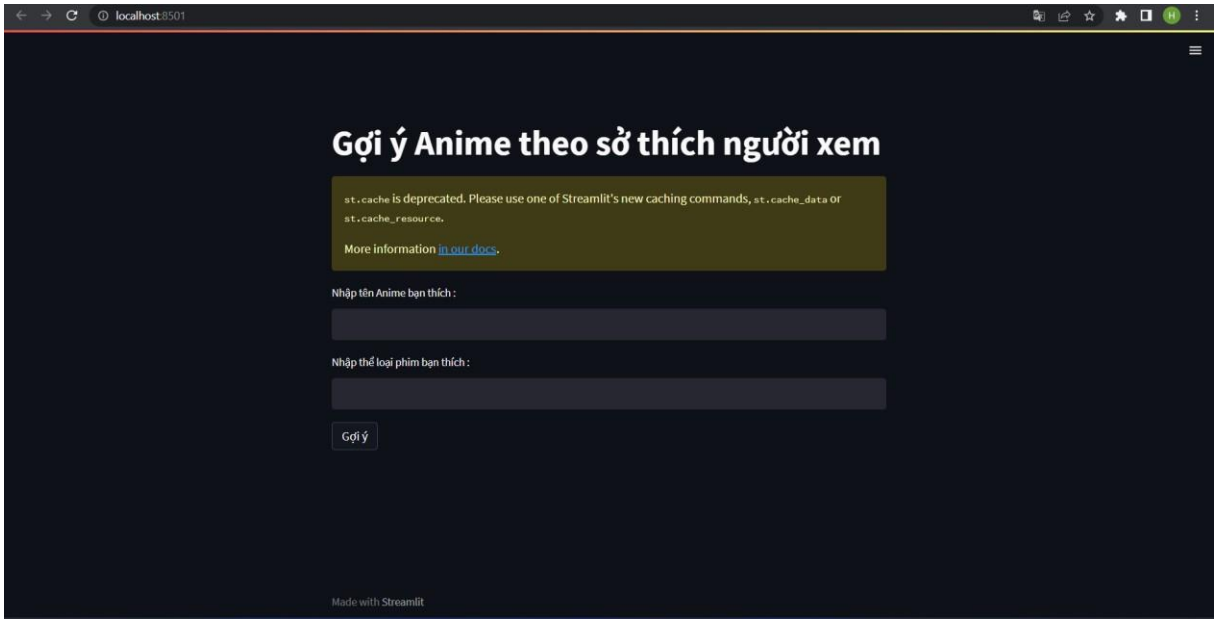
CHƯƠNG IV. THỰC NGHIỆM VÀ KẾT LUẬN

4.1. Chạy chương trình

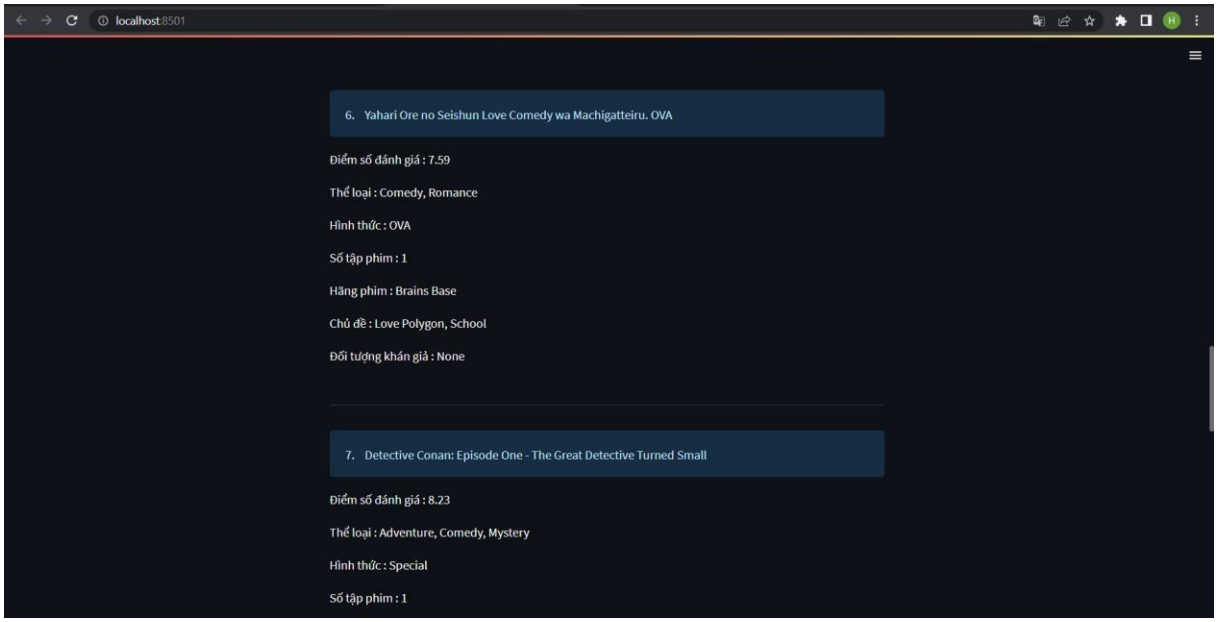
Ta mở Terminal của Visual Studio Code, gõ vào dòng lệnh sau:

Sau khi chạy dòng lệnh trên, chương trình sẽ tạo một localhost để truy cập vào trang web gợi ý anime.





Ta nhập vào tên phim anime và thể loại mà bạn muốn gợi ý. Sau khi nhập xong, ta bấm nút "Gợi ý". Chương trình sẽ hiện ra danh sách gồm tên các phim và thông tin của chúng.



4.2. Kết quả đã đạt được và hạn chế

Đồ án **"Gợi ý Anime theo sở thích người xem"** là đề tài có tính ứng dụng cao trong đời sống. Qua việc nghiên cứu và hoạn thiện đồ án này, em đã có thêm kiến thức về xử lý dữ liệu thô, cách làm việc với dữ liệu một cách khoa học, lấy dữ liệu thực tế trên web và tạo ra một chương trình gợi ý (Recommendation System) đơn giản.

Bên cạnh những kết quả đã đạt được, chương trình vẫn còn những điểm hạn chế. Vì lấy dữ liệu thực tế mất rất nhiều thời gian nên số lượng dữ liệu còn có hạn. Giao diện web của chương trình vẫn chưa được đẹp mắt. Thuật toán gợi ý phim vẫn còn khá sơ sài khi so với thuật toán dùng trong thực tế.

4.3. Kết luận

1. **Xử lý dữ liệu thô:** Em đã học cách làm sạch, tiền xử lý và chuẩn bị dữ liệu để có thể sử dụng trong mô hình gợi ý. Điều này bao gồm việc xử lý các giá trị thiếu, loại bỏ dữ liệu nhiễu và chuẩn hóa dữ liệu.

2. **Làm việc với dữ liệu một cách khoa học:** Em đã hiểu và thực hành các bước cần thiết để phân tích và trực quan hóa dữ liệu, từ đó rút ra những thông tin hữu ích và phục vụ cho việc xây dựng mô hình gợi ý.

3. **Lấy dữ liệu thực tế trên web:** Em đã nắm vững kỹ thuật web scraping để thu thập dữ liệu từ các nguồn trực tuyến, một kỹ năng quan trọng trong việc xây dựng cơ sở dữ liệu thực tế cho các dự án dữ liệu.

4. **Tạo ra một chương trình gợi ý đơn giản:** Em đã áp dụng các thuật toán gợi ý, như lọc cộng tác và lọc dựa trên nội dung, để xây dựng một hệ thống gợi ý anime cá nhân hóa cho người dùng. Qua đó, em đã nắm bắt được quy trình xây dựng và đánh giá một hệ thống gợi ý hiệu quả.

Tóm lại, đồ án này không chỉ giúp em mở rộng kiến thức và kỹ năng trong lĩnh vực xử lý và phân tích dữ liệu mà còn cung cấp một ứng dụng thực tiễn có thể giúp người dùng khám phá các bộ anime phù hợp với sở thích của họ. Em rất tự hào về những gì đã đạt được và mong muốn tiếp tục phát triển những kiến thức và kỹ năng này trong các dự án tương lai.

Trong quá trình làm đồ án thực tập chuyên ngành, rất khó để tránh khỏi các sai sót không đáng có. Em xin cảm ơn thầy **Đỗ Duy Cốp** và các bạn đã giúp đỡ em trong quá trình nghiên cứu và hoàn thành đồ án.

TÀI LIỆU THAM KHẢO

https://scikit-learn.org/stable/

https://www.dictionary4it.com/term/CountVectorizer-7850/

https://www.geeksforgeeks.org/cosine-similarity/

https://www.youtube.com/watch?v=hM8UdUt1PqU

https://myanimelist.net/